

**Programación I**  
**Grado de Ingeniería Informática y Doble Grado II-ADE - ETSIINF - UPM**  
**SOLUCIONES EXAMEN EVAL 1**  
**28-oct-2019**

**Ejercicio 1 (2 puntos)**

// PRE: direccion pertenece al conjunto {'N','S','E','O'}

**solución 1** ( empleando if else)

```
static String giro90 ( char rumbo) {  
    String nuevoRumbo="";  
    if ( rumbo == 'N')  
        nuevoRumbo= "Este";  
    else if ( rumbo == 'E')  
        nuevoRumbo= "Sur";  
    else if ( rumbo == 'S')  
        nuevoRumbo= "Oeste";  
    else  
        nuevoRumbo= "Norte";  
    return nuevoRumbo;  
}
```

**solución 2**( empleando switch)

```
static String giro90 ( char rumbo) {  
    String nuevoRumbo = "";  
    switch (rumbo) {  
        case 'N': nuevoRumbo = "Este" ; break;  
        case 'E': nuevoRumbo = "Sur" ; break;  
        case 'S': nuevoRumbo = "Oeste" ; break;  
        default: nuevoRumbo = "Norte" ;  
    }  
    return nuevoRumbo;  
}
```

**solución 3**( empleando switch)

```
static String giro90 ( char rumbo) {  
    switch (rumbo) {  
        case 'N': return "Este";  
        case 'E': return "Sur";  
        case 'S': return "Oeste";  
        default: return "Norte";  
    }  
}
```

## Ejercicio 2 (3 puntos)

2.1  $f(4, 7) \rightarrow 2$

| inicio           | a | b | e     | n |
|------------------|---|---|-------|---|
|                  | 4 | 7 | false | 4 |
|                  | 4 | 7 | false | 5 |
|                  | 4 | 7 | true  | 6 |
| return $6-4 = 2$ |   |   |       |   |

2.2  $f(14, 25) \rightarrow 4$

| inicio             | a  | b  | e     | n  |
|--------------------|----|----|-------|----|
|                    | 14 | 25 | false | 14 |
|                    | 14 | 25 | false | 15 |
|                    | 14 | 25 | false | 16 |
|                    | 14 | 25 | false | 17 |
|                    | 14 | 25 | true  | 18 |
| return $18-14 = 4$ |    |    |       |    |

2.3 Se acepta cualquiera de las siguientes suposiciones:

### solución 1

PRE: existe primo en  $[a, b]$

Devuelve el posición del primer primo en  $[a, b]$ .

### solución 2

PRE: cierto

En caso de ser  $a > b$  devuelve 0.

En otro caso si no existe ningún número primo en  $[a, b]$  se devuelve  $b-a+1$  (la posición de  $b$ ).

En caso de existir algún número primo en el intervalo  $[a, b]$  devuelve la posición del primero de ellos.

### Ejercicio 3 (4 puntos)

#### 3.1

##### **solución 1**(usando bucle while)

```
// PRE: n>=0
// POST: resultado = 10^(nºDigitos(n))
static int magnitudSuperior (int n) {
    int resultado = 10;
    int i = n;
    while (i > 9)
    {
        resultado = resultado * 10;
        i = i / 10;
    }
    return resultado;
}
```

##### **solución 2** (usando bucle for)

```
static int magnitudSuperior ( int n) {
    int resultado = 10;
    for (int i=n; i>9; i=i/10) {
        resultado= resultado*10;
    }
    return resultado;
}
```

##### **solución 3** (usando recursión)

```
static int magnitudSuperior (int n) {
    if (n<10)
        return 10;
    else
        return 10*magnitudSuperior(n/10);
}
```

#### 3.2

##### **solución 1**

```
System.out.println("magnitudSuperior ( 20345 )=" + magnitudSuperior ( 20345 ));
System.out.println("agnitudSuperior ( 1000000 )=" + magnitudSuperior ( 1000000 ));
```

##### **solución 2**

```
boolean prueba1 = magnitudSuperior ( 20345 ) == 100000;
boolean prueba2 = magnitudSuperior ( 1000000 ) == 10000000;
```

#### Ejercicio 4 (1 punto)

##### **Solución 1** (usando la clase Math)

```
static int concatenar ( int a, int b) {  
    int num_digitos = (int)Math.log10(b) + 1;  
    return a*(int)Math.pow(10,num_digitos) + b;  
}
```

##### **Solución 2** (invocando magnitudSuperior)

```
static int concatenar ( int a, int b) {  
    return a*magnitudSuperior(b) + b;  
}
```